Machine Comprehension using Bidirectional LSTM *

Aman Kumar, B00777628 Asok Kalidass Kalisamy, B00763356
Balaji Dhakshinamoorthy, B00777437
Viswanath MuthuKumaraSwamy Sathananth, B00782640

May 22, 2018

1 Abstract

Machine Comprehension, an information retrieval task is one of the main problems in Natural Language Processing. The task is to answer questions based on a passage and it involves some intricate semantic interaction between a question and the passage. Stanford Question Answer Dataset (SQuAD) dataset introduced recently produced more challenges in terms of semantic interpretation of passage based on a query and the answers are not from a small set of candidate answers but they are crowd-sourced answers of varying lengths. In this project, we use a bidirectional LSTM model along with the attention mechanism based on cosine similarity to solve this problem. We propose pooling technique on top of attention mechanism so that our model extracts high-level information and achieves a better state of art result in lesser time than previously proposed LSTM approaches.

Keywords: Machine Comprehension, SQuAD Dataset, Bidirectional LSTM, Attention Mechanism.

2 Introduction

One of the most challenging tasks in Natural Language Processing is to make a machine understand the questions asked by a human being and find the answer for them. Machine Comprehension task makes a system understand the context in natural language and answer a wide variety of complex questions posed by human beings. How to make the machine to comprehend a passage and answer challenging questions? is the notion of this project.

The introduction of SQuAD Dataset by Rajpurkar et al. [1] has passages created through crowdsourcing with several questions on each passage. More importantly, answers for the passage can be a single word or a long phrase. The need of a dataset for more complex reasoning was fulfilled by introducing more challenging questions which required more semantic interaction between the questions and passage. The SQuAD dataset made Machine Comprehension a Textual Entailment problem, where the question can be treated as a text and the span answers with varying lengths can be treated as a hypothesis for that question. The Dataset has comparatively more magnitude in terms of the size of the data and more challenging questions than the previous dataset and it becoming a popular dataset for research related to the Machine Comprehension problem in the recent times.

Another key reason behind the advancement in this domain is an end to end Neural Network model. Most successful models have two things in common; a Recurrent Neural Network model to process sequential inputs and an attention model to handle the interaction between question and passage. This model needs more time for inference particularly for longer text because of its recurrent nature and attention mechanism.

The motivation behind this project is to design a system with bidirectional LSTM and better attention mechanism to obtain a trade-off between model accuracy and computation time. The advent of Neural Networks has to lead to an increase in interest towards machine comprehension

*Submitted as a part of course work CSCI - 6509 Winter 2018, Computer Science, Dalhousie University.

problem (Hill et al.[14]; Yin et al. [13]). However, they depend on an earlier dataset with single candidate answers that make them unsuitable for SQuAD. But recently there is a good number of papers published based on bidirectional LSTM (Long Short-Term Memory) and they produced better results than traditional NLP approaches.

We propose a model which process inputs using bidirectional LSTM similar to the approach proposed in recent publications. And we propose a novel attention technique that computes the cosine similarity between the results of question and passage LSTM layers and applies maximum and average pooling techniques on top of them to predict the start and end indices of the answer in the paragraph. This introduction of pooling concepts makes the model better through high-level information extraction and the with an addition of further stacked LSTM layers on top of them produces better answer prediction in efficient time.

In this report, we have listed out all the techniques and implementation process for Machine Comprehension. The outline of these reports goes as follows: Section 3 describes the past related methodology and techniques used to develop the machine comprehension techniques; Section 4 is the problem statement and our methodology; Section 5 describes the various implementation phases; Section 6 is the conclusion of the report; Section 7 provides the future work.

3 Related Work

MCTest Dataset created by Richardson et al. [16] which contains a set of stories and associated questions was used extensively in Machine Comprehension (MC) research. Bingning wang et al. [2] proposed the model based on textual entailment technique which combines question and candidate answer and treats as one sentence and checks for the semantic similarity between that sentence to all another sentence in the document by using recurrent neural networks for answer prediction. Their approach was later improved by Hai Wang et al. [3] by using the syntax, frame semantics, word embeddings of the document to predict the candidate answer. The previous techniques was found to be inefficient, as the answer span can be of varying lengths, after release of SQuAD dataset by Rajpurkar et al [1] where they built the model to collect the candidate sentences from passage through various NLP technique like dependency parser, named entity recogniser and applied the feature engineering technique along with logistic regression to rank the candidate sentence. However, few of the question answers don't have matching answers with that predicted candidate sentence.

After the release of train and dev SQuAD dataset to the developer community, there was phenomenal development in the field of MC, numerous neural networks techniques were adopted to build the model with good accuracy rate. At present, the Human Performance model by Rajpurkar et al. [1] tops the SQuAD leaderboard with an impressive F1 score of 91.221 [12].

Zhiguo wang [4] introduced the Bidirectional LSTM with special attention mechanism by matching contextual embedding of passage with the question from multiple perspectives for predicting start and end indices of the answer. But this model takes more time for training than the rest of the works. Shouhang Wang et al. [5] used a combination of match LSTM and Pointer Answer technique [6] for predicting the sequence of positions from the passage as the answer. But however, the sequence need not be consecutive. So he further modified by predicting the start and end indices as an answer. Kenton Lee et al. [7] further optimized the early model by enumerating all possible answer spans from the LSTM model and applied novel technique based on the recurrent computation of shared substructures for generating all possible answer spans and ranking them based on their similarity.

Our work is related to Minjoon Seol et al. [8], where the authors used the attention mechanism weights based on passage words and query words along with LSTM Layers. The same approach was followed by Adams Wei Yu et al. [9] but by using Convolutional Neural Networks (CNN). We are planning to use the cosine similarity between passage and query words for calculating weights based on Maximum Pooling and Average Pooling attention mechanism techniques along with Bidirectional LSTM to generate the start and end indices of the candidate answer. The attention mechanism techniques will increase the performance of the model as we are considering the maximum and average weights instead of entire weights as attention.

4 Problem Definition and Methodology

Machine comprehension task involves a question and a supporting passage. The task is to come up with an answer from the passage based on the question. For better performance, the model should understand the question, reason it among the passage and should predict the correct answer span.

There was a need for a comprehensive dataset and more semantically interpretable model. Large datasets play a crucial role in designing effective systems. Several datasets were proposed to solve the machine comprehension problem, few of them are

4.1 CBT Cloze Dataset

The children book test cloze dataset is created using 500 fictional passages and multiple choice type questions based on those passages by Hill et al. [14]. The dataset doesn't pose strong machine comprehension challenges because of their restricted candidate answer choices.

4.2 Cloze Dataset

The cloze dataset is derived mostly from CNN/Daily news articles. Hermann et al. [17] designed a Cloze type questions by leaving out space in the summaries of the news articles. The dataset is huge but it requires less reasoning. Often, it is possible to predict the missing word based on the knowledge of the structure of the articles.

4.3 MCTest Dataset

The MCTest Dataset Consists of 660 stories created through crowdsourcing, with 4 questions for each story and 4 answer choices for each question. This dataset requires some logical reasoning and is used to evaluate the machine comprehension problem. The main drawback of this dataset is that it doesn't have answers of varying length. This reduces the amount of reasoning involved in getting to the answer.

4.4 SQuAD Dataset

The SQuAD dataset is usually represented as a set of three tuples (a passage, question, and answers). Where a passage is represented in the form of words like p1,p2,...pn. and for each passage, there are several sets of questions each having representation in the form of words like q1,q2,...qn. In our Approach answer is usually represented in form of A=A start, A end.where Astart denotes the start index and Aend denotes the end index of the answer span to look for in the passage. such that 1<=Astart<=Aend<=N where N denotes the passage word size. So The Machine Comprehension task is to design a reasoning model to predict the answers in the form of Astart and Aend. So Astart can be stated as a conditional probability of finding it, given the question and passage as input and Aend can be computed as finding the conditional probability of Aend given passage, question, and Astart as input.

4.5 Bidirectional LSTM and Attention Mechanism model

In this section, we propose a Model with Bidirectional LSTM and Specific Attention Mechanism to predict the probability distributions of answer indices Astart and Aend. In the below Figure 1 shows our model representation. The last layer denotes the answer prediction stage.The prediction happens based on the probability distributions of start and end indices. And other layers beneath it comprises of the information flow between question and passage. Given a question and a passage, our model works through the following phases.

4.5.1 Input Phase

In this phase, words of both passage and question are processed separately. The words should be represented in the form of vectors. Because of sparse data problem, one hot representation is not an ideal choice for this problem. So, we have used the GloVe pre-embedding technique probosed by Pennington et al. [11] for this task. This technique converts word into a d dimensional word vector based on its similarity score with the pre-trained GloVe word vectors. In that procedure, all the words from passage and query get converted to 'd' dimensional word vectors and it will be passed as an input to the subsequent LSTM Layers.

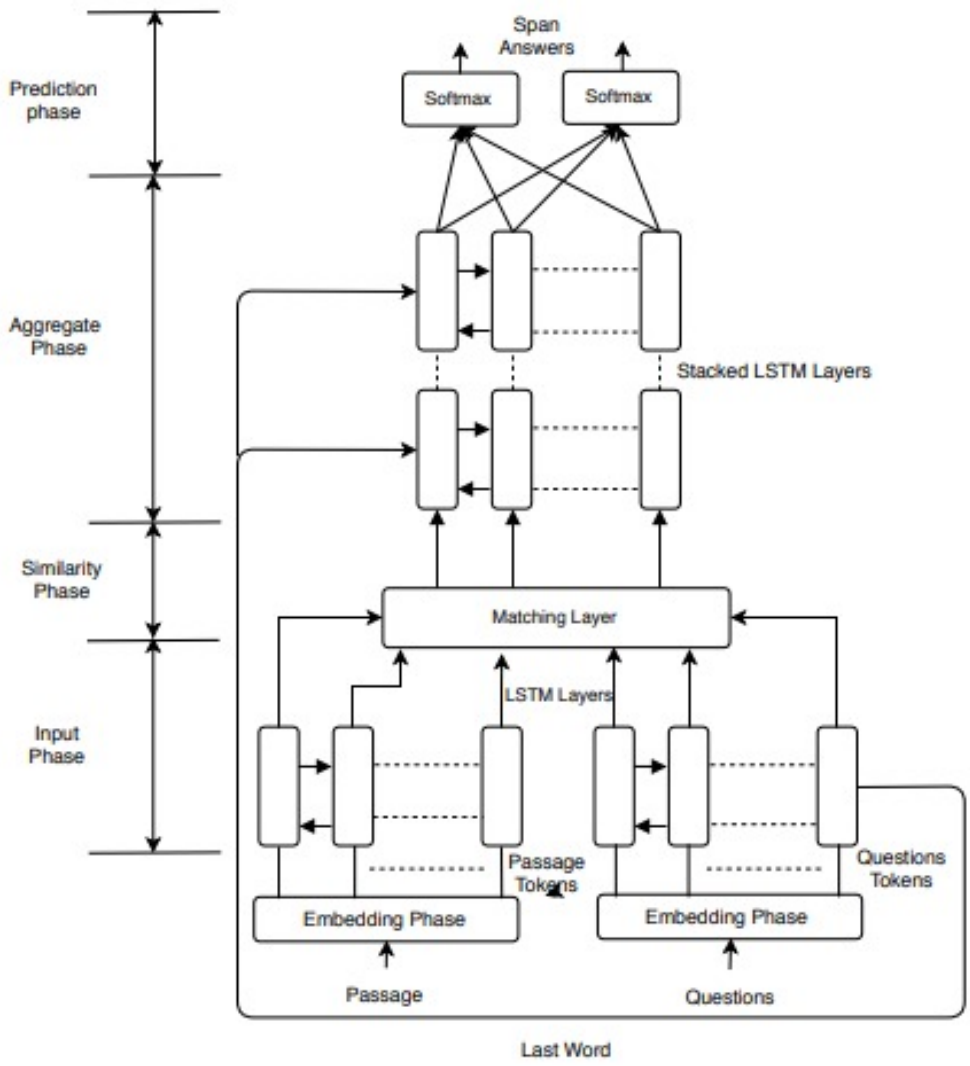

Figure 1: *Model representation of Machine Comprehension - Drawn using [10].*

The passage words p1, p2,... pn are changed to a 300-dimensional word vector, similarly, the question words q1,q2,... qm gets converted into a 300-dimensional word vector. Following this step, the passage word vectors are passed as an input to the Bidirectional LSTM. The hidden layer size is directly proportional to the number of tokens in the passage. Similarly, for the questions, a separate Bidirectional LSTM layer is created with hidden layer size as the number of word vectors in question. Bidirectional LSTM is designed in such a way that output of each hidden layer knows about the word interaction both from the future and past words. The outputs of the LSTM layers of both question and passage are passed as input to the matching phase.

Let the input of the each LSTM in the form of a word vector sequence be [q1,q2, ...,qm] for a question with m word tokens. Similarly let the passage tokens be [p1,p2,...,pn] for a passage with word token respectively. Let their corresponding output be of the form [Hq1, Hq2,... Hqm] and [Hp1,Hp2,...,Hpn].

$$\overleftarrow{HQ_i} = BiLSTM(\overleftarrow{q}_i) \quad i = 1, \ldots, M$$
$$\overrightarrow{HQ_i} = BiLSTM(\overrightarrow{q}_i) \quad i = 1, \ldots, M$$

And for the passage:

$$\overleftarrow{HP_i} = BiLSTM(\overleftarrow{p}_i) \quad i = 1, \ldots, N$$
$$\overrightarrow{HP_i} = BiLSTM(\overrightarrow{p}_i) \quad i = 1, \ldots, N$$

4.5.2 Matching Phase

This is the main layer of our model. The main goal is to compute the cosine similarity between the output words of the passage and query words and then apply the pooling technique to extract information from them.

Cosine similarity is applied to the output of Bidirectional LSTM layers for the question and the passage vectors. Cosine similarity between the passage and query vectors is calculated using the following formulae,

$$\overleftarrow{T_{i,j}} = \overleftarrow{Hq_i}^T.\overleftarrow{Hp_j}/||\overleftarrow{Hq_i}||.||\overleftarrow{Hp_j}||Hp_j/||\overleftarrow{Hq_i}||.||\overleftarrow{Hp_j}||$$
$$\vec{T_{i,j}} = \vec{Hq_i}^T.\vec{Hp_j}/||\vec{Hq_i}||.||\vec{Hp_j}||Hp_j/||\vec{Hq_i}||.||\vec{Hp_j}||$$

where i=1,. . . ,M and j = 1,. . . ,N

The result is a matrix with question LSTM output as the rows and passage LSTM output as the columns. Each cell in the matrix denotes the cosine similarity between a question LSTM output word and the corresponding passage LSTM output word. Since each passage word column has a similarity score of all question words, there is a lot of dimension information to be processed by the further layers. Our goal is to get a high dimensional summary information for each passage word to compute the answer spans. To achieve this and also to make further processing faster, we introduce two pooling layers. These layers assign a similarity score for each passage word depending on how similar it is to question words.

4.5.3 Max pooling

Max pooling fetches the maximum value in a given area and returns that as output. In our case, each passage word will be passed through a max pooling layer to fetch the maximum cosine similarity of that passage word with all of the question words. This can be represented as follows,

$$\vec{f_i} = max_j \vec{P_i} * \vec{Q_j}$$
$$\overleftarrow{f_i} = max_j \overleftarrow{P_i} * \overleftarrow{Q_j}$$

Where i=1,. . . m denotes the words in question.

4.5.4 Average Pooling

Average pooling is a concept which computes a mean of all value in the area and returns that as a result. In this case, it returns the mean of all cosine similarity score of each passage word. It is represented as,

$$\vec{f_i} = avg_j \vec{HP_i} * \vec{HQ_j}$$
$$\overleftarrow{f_i} = avg_j \overleftarrow{HP_i} * \overleftarrow{HQ_j}$$

Where i=1,. . . ,m denote the words in the question.

The outputs of the max pooling and average pooling layer will then be concatenated and passed as an input to the further layers.

4.5.5 Aggregate Phase

This phase takes the output of the previous matching phase for all passage words and applies that as an input to the further stacked Bidirectional LSTM layers. This stacked LSTM layers is used to find the word interaction between the passage words which has the context of question words. Through this function, more complex semantic interpretation can be computed. The output of this LSTM layers will then be passed to the final layer.

4.5.6 Prediction Layer

This is the final layer of our model. The output of the previous aggregated vector at each time step will be passed to the two feed-forward neural network with dense of 1 so that each passage word will have final probability distribution value. The softmax operation will be performed on the first network to find the start indices. Then this information will be passed to second feed forward network and the softmax operation will then be performed to predict the end index of the answer span. Once the start and end index is generated corresponding answer span from the passage will be searched and returned as an answer to the question. This summarizes all the information flow happening in our model.

5 Experiment Design

We constructed our model with the help of SQuAD Dataset. The dataset contains 87599 training samples and 10570 validation samples. Because of resource constraints, we considered only small number as a training instance for our model. For Neural Network construction we considered Keras a high-level API with a tensor flow as backend. We used a many to many models of Bidirectional LSTM with an equal number of inputs and outputs. Hidden layer size of each Bidirectional LSTM is same as the number of input tokens. We apply dropout to each layer with the ratio set to 0.2. At the moment, we haven't set up any momentum based optimizer but in future, We will extend the model with Adam based optimizer proposed by Kingma et al. [15] in order to minimize Cross Entropy loss.

5.1 Implementation Phases

Implementation of this project happens in following multiple phases. Please refer the Figure 2 for all the implementation phases.

5.1.1 Tokenization Phase

We use the similarity between question and the context(the paragraph) to get to the answer. To start with, the context and the respective questions from the SQuAD Dataset are read into memory and split into separate bags of words. To avoid problems due to Unicode, only the words in GloVe vector are considered.

5.1.2 Word Embedding Phase

To perform word embedding for each word, we make use of Global Vectors for Word Representation (GloVe). GloVe provides the co-occurrence statistics of a given corpus. GloVe by itself is an unsupervised learning algorithm. We can either run GloVe on our own corpus or download one of the pre-trained word vectors. The pre-trained versions of GloVe vector were run on Wikipedia 2014 pages and Gigaword 5. Commonly, GloVe vectors have dimension 50,100,200,300. More the number of dimensions, better is the result and more is the time taken. The benefit of using GloVe is that it implements a skip gram model along with matrix factorization which makes it statistically strong. The output of this phase is an embedded matrix for words in both the question and the context.

5.1.3 First Level LSTMs

The weighted matrix produced by performing word embedding on the words in both question and the context is passed to separate LSTMs. The number of nodes in the LSTM is determined by the number of tokens in question and the answer respectively. These LSTMs are many to many. They take in n words of 100 dimensions each (dimension depends on the GloVe dataset chosen). The output of this layer is 'n' weights.

Other phases like the Cosine Similarity and Stacked LSTM layers couldn't be implemented because of resource constraints.

5.2 Resource Constraints

The model we proposed is quite computationally intensive. The earlier works on these models used NVIDIA Pascal 100 to run these tasks. Since we don't have any machine with GPU, we were not

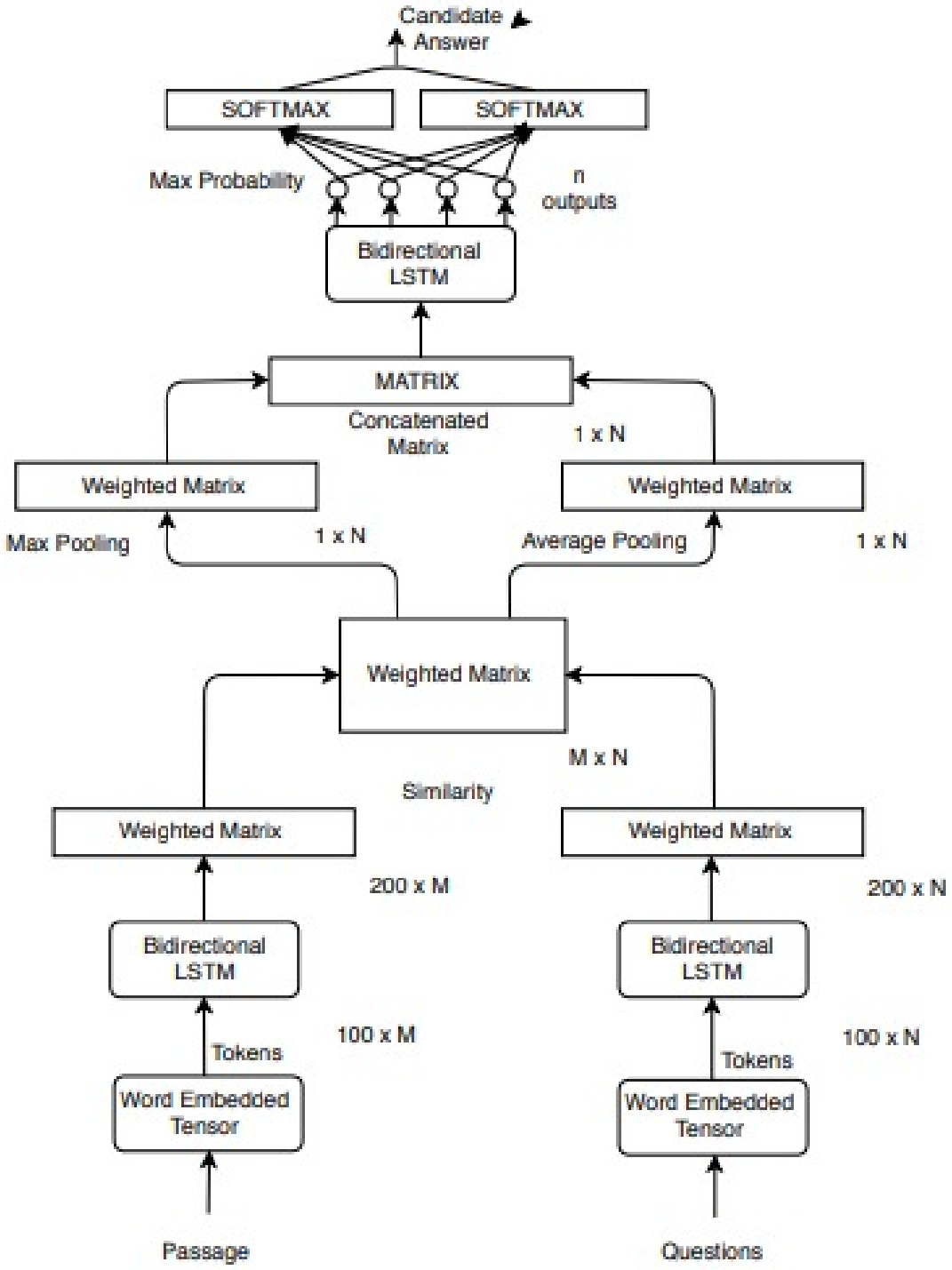

Figure 2: *Implementation Flow Diagram - Drawn using [10].*

able to construct the model properly. We had to reduce the number of training instances and word embedding dimensions. As a result of which the quality of our results went down.

With respect to the results, we have constructed our model on Keras in python language. The source code is provided to show our work. We are facing some issues in implementing our matching model that is based on cosine similarity. We have constructed a multi-level LSTM using Stacked LSTM in Keras. But we couldn't complete a part of the work that involves cosine similarity phase. The cosine similarity phase needs a powerful GPU resource to estimate the efficiency of our model with other works based on SQUAD Dataset.

6 Conclusion

In this work, we have proposed a model using Bidirectional LSTM with attention mechanism based on cosine similarity as our solution for Machine Comprehension. We constructed the model using pooling techniques to improve the efficiency of information extraction. In future after evaluation, we will try to add more high-level information extraction technique in the matching phase along with pooling technique to further optimize the model so that our model achieves a better result on the SQUAD Dataset leaderboard.

7 Future Work

In the time allotted for the project, we were able to achieve only a part of what we had actually planned due to various constraints. We are trying to implement maximum pooling and average pooling approaches. Our immediate plan would be to finish setting this up and test it against the existing models. We also plan to submit our model for evaluation by the SQuAD team. We also plan to make this search more accurate by using algorithms like Beam Search to get better results. We are also planning to explore more into the region of Semantic Search. As described already more the amount of context understood by the system, better are the results and semantics play a major role in understanding the concept. We believe some of these search methods can be helpful in finding a better solution to this problem.

8 References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. ICLR, 2016. [Accessed Mar. 15, 2018].

[2] Bingning Wang, Shangmin Guo, Kang Liu, Shizhu He, and Jun Zhao. Employing external rich knowledge for machine comprehension. ICLR, 2017. [Accessed Mar. 15, 2018].

[3] Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. Machine comprehension with syntax, frames and semantics. [Accessed Mar. 15, 2018].

[4] Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. Multi-perspective context matching for machine comprehension. ICLR, 2016. [Accessed Mar. 15, 2018].

[5] Shuohang Wang and Jing Jiang. Machine comprehension using match-LSTM and answer pointer. ICLR, 2017. [Accessed Mar. 15, 2018].

[6] Oriol Vinalys, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In proceedings of conference on Advances in Neural Information Processing Systems, 2015., 2015. [Accessed Mar. 15, 2018].

[7] Kenton Lee†, Shimi* Salant†, Tom Kwiatkowksi‡, Ankur Parikh‡, Dipanjan Das‡, and Jonathan Berant*. Span: Learning recurrent span representations for extractive question answering. 2017. [Accessed Mar. 15, 2018].

[8] Minjoon Seo1, Aniruddha Kembhavi, and Hananneh Farhadi, Ali Hajishirzi. Bidirectional attention flow for machine comprehension. ICLR 2017, 2017. [Accessed Mar. 15, 2018].

[9] Adams Wei Yu∗, David Dohan2†, Minh-Thang Luong2†, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Span: Fast and accurate reading comprehension by combining self-attention and convolution. Published as a conference paper at ICLR 2018, 2018. [Accessed Mar. 15, 2018].

[10] "free flowchart maker and diagrams online," Flowchart Maker & Online Diagram Software. [Online]. Available: https://www.draw.io/. [Accessed: 15-Mar-2018].

[11] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing

(EMNLP), 2014. [Accessed Mar. 15, 2018].

[12] "SQuAD," The Stanford Question Answering Dataset. [Online].
Available: https://rajpurkar.github.io/SQuAD-explorer/. [Accessed: 15-Apr-2018].

[13] W. Yin, S. Ebert, and H. Schütze, "Attention-Based Convolutional Neural Network for Machine Comprehension," Proceedings of the Workshop on Human-Computer Question Answering, 2016. [Accessed: 15-Apr-2018].

[14] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The Goldilocks principle: Reading children's books with explicit memory representations. In proceedings of the International Conference on Learning Representations, 2016. [Accessed: 15-Apr-2018].

[15] Diederik P. Kingma, and Jimmy Lei Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION," Published as a conference paper at ICLR 2015, 2015 [Accessed Mar. 15, 2018].

[16] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw, "MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text," Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013. [Accessed: 15-Apr-2018].

[17] Karl Moritz Hermann, Toma´s Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom, "Teaching Machines to Read and Comprehend," Proceedings of the Neural Information Processing Systems Conference, 2016. [Accessed: 15-Apr-2018].